Visualization of Large Molecular Trajectories

David Duran, Pedro Hermosilla, Timo Ropinski, Barbora Kozlíková, Àlvar Vinacua, and Pere-Pau Vázquez

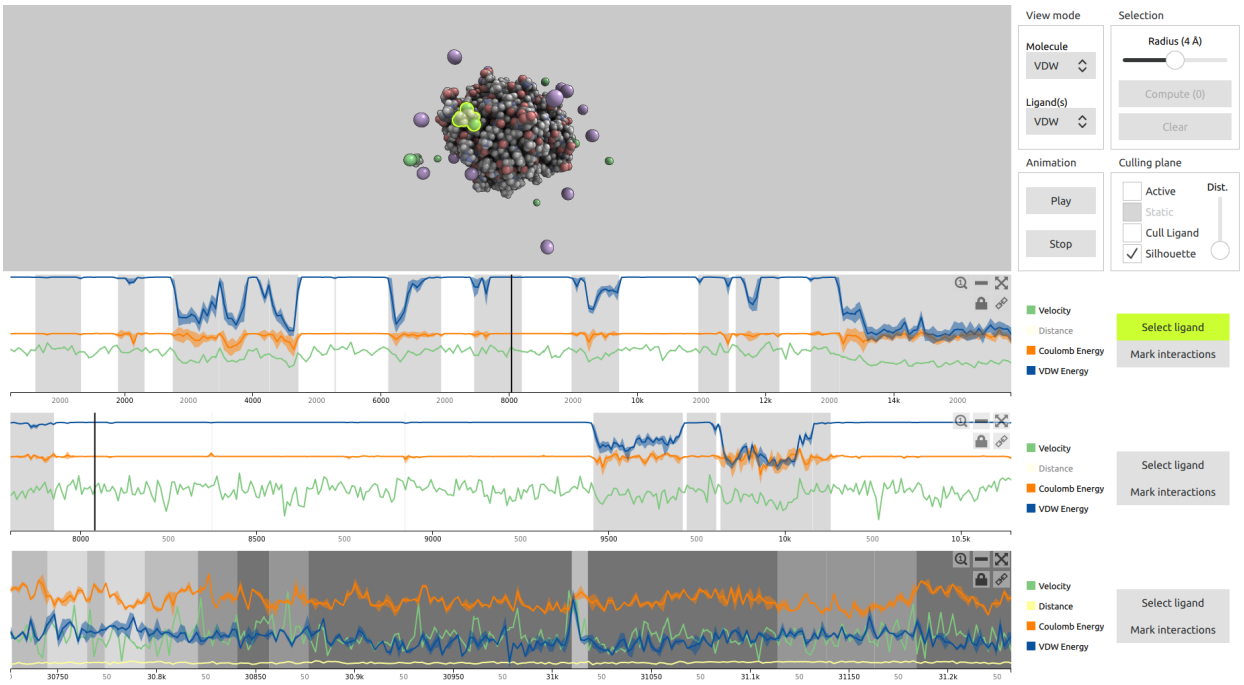


Fig. 1. Overview of the system: The 3D exploration of the molecular trajectory appears on top of the enhanced charts of three different ligands involved in a simulation. The plots are linked bidirectionally: researchers can perform selections in 2D or 3D to see relevant information highlighted in the other view. A 3D selection, for instance, will highlight the intervals of the 2D plots where the ligand interacts with the selected region.

Abstract—The analysis of protein-ligand interactions is a time-intensive task. Researchers have to analyze multiple physico-chemical properties of the protein at once and combine them to derive conclusions about the protein-ligand interplay. Typically, several charts are inspected, and 3D animations can be played side-by-side to obtain a deeper understanding of the data. With the advances in simulation techniques, larger and larger datasets are available, with up to hundreds of thousands of steps. Unfortunately, such large trajectories are very difficult to investigate with traditional approaches. Therefore, the need for special tools that facilitate inspection of these large trajectories becomes substantial. In this paper, we present a novel system for visual exploration of very large trajectories in an interactive and user-friendly way. Several visualization motifs are automatically derived from the data to give the user the information about interactions between protein and ligand. Our system offers specialized widgets to ease and accelerate data inspection and navigation to interesting parts of the simulation. The system is suitable also for simulations where multiple ligands are involved. We have tested the usefulness of our tool on a set of datasets obtained from protein engineers, and we describe the expert feedback.

Index Terms—Molecular visualization, simulation inspection, long trajectories

1 Introduction

- *David Duran is with the ViRVIG Group, UPC Barcelona. E-mail: dduran@cs.upc.edu.*
- *Pedro Hermosilla is with the Visual Computing Group, U. Ulm. E-mail: pedro-1.hermosilla-casajus@uni-ulm.de.*
- *Timo Ropinski is with the Visual Computing Group, U. Ulm. E-mail: timo.ropinski@uni-ulm.de.*
- *Barbora Kozlíková is with the Masaryk University. E-mail: kozlikova@fi.muni.cz.*
- *Àlvar Vinacua is with the ViRVIG Group, UPC Barcelona. E-mail: alvar@cs.upc.edu.*
- *Pere-Pau Vázquez is with the ViRVIG Group, UPC Barcelona. E-mail: pere.pau@cs.upc.edu.*

Molecular dynamics simulations [8, 39] are computer simulations of the physical movements of atoms and molecules, and the interactions between them. These simulations are used in several areas, such as chemical physics, materials science, and modeling of biomolecules. In pharmacology, drug design, and enzymatic catalysis, molecular dynamics simulations predict the binding mode and binding affinity of a small molecule (the drug) with a biomolecule. The advances in hardware, such as the Anton machine [37], a device specially crafted to compute molecular simulations, and software, e.g., new simulation models, such as the Markov State Model (MSM) [7], have significantly increased the amount of data to analyze [5, 20]. Unfortunately drug design still largely depends on human input to analyze the outcomes of the simulations and discuss the potential modifications to drugs to

 DOI 10.1109/TVCG.2018.2864851

make them effective. While the analysis of the simulation data by itself is a complex problem, due to the many variables at work, it can even grow worse when simulations involve thousands of snapshots (time steps). Currently, no tools exist that facilitate the analysis of very large trajectories, so the consequent back and forth chart inspection or zoom-in and -out become tedious and extremely time consuming.

Our approach is tailored to deal with very large trajectories (from thousands to hundreds of thousands of steps) that may involve several ligands simultaneously. This poses several challenges: *i)* dealing with the data itself, that amount to gigabytes of memory, *ii)* finding the adequate representation that visually highlights important features on the whole simulation and guides the user towards the most interesting parts of the simulation, and *iii)* providing interaction techniques that facilitate instant, progressive exploration of the data. To address these tasks, we have created a new tool built specifically for the inspection of large trajectories. The key design requirement of the proposed system was to integrate spatial and non-spatial information by means of interaction and visualization. To do so, we have employed the existing 2D and 3D visualization techniques and have interlinked them through a hybrid interaction scheme in such a way that the experts who are familiar with these representations can concert them effortlessly. In this way, we can meet the main requirements of a trajectory exploration system: domain experts can use energy charts to obtain an overview of the states of the simulation by showing potentially relevant simulation steps, while at the same time being able to perform a 3D exploration to obtain a deeper understanding of the protein-ligand interplay. Thus, we let the user inspect the simulation via bidirectional linking from the 2D charts to 3D simulation and vice versa. Based on the simulation data, we automatically detect regions of potential interest and represent this information within specialized widgets that the user may click on to quickly jump to these parts of the simulation, while the detailed inspection can then be performed within the updated 3D view. Conversely, the user can select the regions of interest in the 3D view, and the system will highlight the zones of the charts where such selected 3D volumes are visited by the ligand. By enhancing and combining proven spatial and non-spatial visualization techniques, we ensure that the developed system can be used by a wide audience of domain experts, without posing new challenges wrt. visualization literacy. As a result, the inspection of extremely long trajectories is greatly facilitated. In summary, the main features of our system are:

- Enhanced trajectory charts that are enriched with the derived information from the trajectories, and that provide visual insights on important ranges of the simulations.
- A set of interaction techniques that facilitate the progressive exploration of multiple charts at once.
- A 3D selection technique that allows the user to pick a 3D region in space and highlights portions of the chart where the ligand visits that region.
- A set of multiple coordinated views with bidirectional linking that facilitate the inspection of multiple ligands at once.

The rest of the paper is organized as follows. First, we introduce related work. In Sect. 3 we present our visualization system. Sect. 4 shows the results and discusses use cases. We discuss the limitations and advantages of our technique with respect to the current software in Sect. 5. Finally, Sect. 6 concludes the paper and discusses future directions of our research in this field.

2 Previous Work

Visualization of biomolecules has enjoyed the interest of researchers for decades as visualization can highly facilitate the process of exploration and understanding of the constitution and behavior of molecules. An overview of traditional as well as novel approaches to visualization of biomolecules can be found in recent surveys [1, 18]. These papers summarize techniques for visualization of static molecules as well as trajectories of molecular dynamics (MD).

Trajectories mostly capture the process of transportation of a small molecule (ligand) to the protein active site, where the mutual reaction between these two molecules can take place. Therefore, the existing approaches to exploration of trajectories are tightly connected with the presence of void space in proteins. This void space can be categorized according to its connectivity with the protein surface and dedicated algorithms for their detection and corresponding visualization methods have already been proposed. A comprehensive overview of the existing algorithms and visualization methods for cavities was recently published by Krone, Kozlikova et al. [19].

Current capabilities of modern GPU cards allow to accelerate the rendering techniques, which opens the possibility of rendering large molecules and whole molecular systems in real-time. Chavent et al. [4] presented methods from computer science and visualization which help biologists to explore large molecular systems. Grottel et al. [14] introduced MegaMol, a prototyping framework for real-time visualization of large particle-based scenes. Another tool for interactive rendering of large scenes containing biomolecular systems, cellVIEW, was published by Le Muzic et al. [22]. However, these tools focus on real-time visualization of large biomolecular systems and scenes and do not deal with the problem of exploring large trajectories.

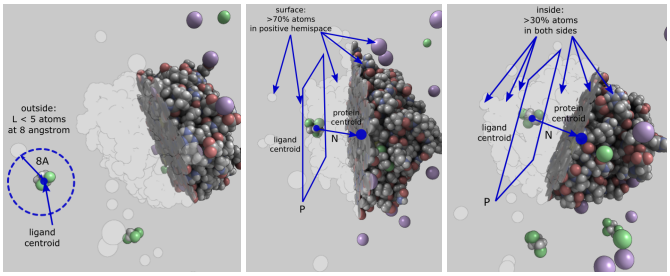
In the area of exploration of protein sequences, several systems have developed specific techniques for gaining insight into the difference between proteins of the same family [27, 31], or discovering features of proteins [32]. In contrast with these techniques, we concentrate on simulation sequences, which may have up to hundreds of thousands of time steps. We need to design specific derived data and interaction techniques to explore them.

MD Trajectories Exploration There are already several existing approaches to visual exploration of protein cavities in trajectories of molecular dynamics. Lindow et al. [24] presented so called dynamic channels whose visual exploration enables the users to analyze the evolution of the cavity over time. Their proposed solution integrates several visualization methods, spanning from static overview representations to animations of trajectories. Byska et al. introduced a method for visual exploration of protein tunnels and the surrounding amino acids over time [3] and a specialized visualization technique for detailed exploration of a tunnel bottleneck and its evolution over time [2]. Nevertheless, none of these approaches considers trajectories containing the ligand movements as well.

Exploration of trajectories, taking into account the ligand interactions with proteins, has been published by Hermosilla et al. [15]. Their tool enables the users to interactively trace the interactions between protein and ligand. Furmanova et al. [12] introduced a system for visual analysis of ligand behavior in large trajectories. It consists of a set of representations enabling the users to identify interesting parts of trajectories, based on the user-defined properties.

Visualization of Large Charts Our work is also partially related to the visualization of large data sets. It is a common problem in Information Visualization, and there are many solutions proposed to many types of data. Commonly, the approaches developed recently have as the main requirement the *interactive exploration* of the data. But the approaches largely differ, depending on the nature of the information as well as the queries that need to be supported [13]. Some approaches deal with discrete data, called events, and the interesting information may come in the form of patterns, outliers, features, etc. The complexity of the data may originate from the variation or its volume [11]. To visualize the whole dataset, several approaches use the aggregation of data in different ways (e.g., [6, 41–43]), and progressive exploration [9, 38, 40]. The focus may be placed on finding the abstract representations or finding patterns, such as repeated sequences of events, and highlight [25] or abstract them, like Malik et al. [30] do with patient data for cohort comparison. Liu et al. [26] visualize large sequences of clicks by a set of motifs that encode consecutive events that appear frequently together in sequences.

In contrast to most of these approaches, our data is two-fold: on one hand the 3D configurations of atoms along the time sequence, and on the other hand the energy values, which are large sets of floating



(a) outside (b) surface (c) inside

Fig. 2. Classifying the ligand position vs the protein: If the ligand interacts with less than 5 atoms of the protein (a), it is assigned to the *outside* class. Otherwise, a plane is created that traverses the centroid of the ligand, with a normal pointing to the centroid of the protein. Then, atoms are classified and the percentage of atoms in each hemispace of the plane determines whether it is classified as *surface* (b) or *inside* (c).

point values. So, we have two different challenges: first to cope with the large amount of information in terms of time steps (snapshots), so we need to design some system that allows an overview representation together with fast inspection of detailed data. Second, we need to provide interactive techniques to quickly pass from the 2D to the 3D rendering and vice-versa, since the 3D view helps researchers to fully understand and complement the information that is displayed by the trajectory charts.

3 Exploration of Long MD Trajectories

In the context of drug design, molecular dynamics simulations are carried out to determine whether certain ligand (drug) can bind to the biomolecule and thus inhibit or activate certain biomolecule function that can be beneficial for the patient. The result of a simulation is a trajectory with information on the positions of the atoms of the participating compounds and on the energy of the system at each step.

Since stable configurations correspond to minima of the energy (binding energy) of the system, the initial and main data presentation that scientists work with are plots of these calculated energy values. The typical approach is to seek for a seemingly interesting portion of the path and then go to the 3D view to inspect the real configuration of the ligand and protein at that point. However, this can be a very time-consuming task, especially in an exploratory phase of very long trajectories (with thousands of snapshots). Our system provides means to accelerate this exploration using two techniques: encoding the potentially interesting regions in the plots themselves, and providing a set of interaction tools for the detailed exploration of the plots and the 3D configuration. Notably, these interactions include the selection of regions in 3D (typically cavities or pockets where scientists want the ligand to bind), and the system determines and visually encodes if and when the ligands enter those selected regions.

The system works as follows:

1. Read the trajectory.
2. Calculate the distances, velocities, and ligand interactions.
3. Calculate the labeling and hierarchical clustering.
4. Interaction.

The trajectory is given in the AMBER format [33], which encodes the individual positions of each of the atoms for each of the steps (called snapshots), together with the energy values for each configuration. From this information, we quickly calculate the positions (distances) of the ligands with respect to the protein, and their velocity at each simulation step, as explained later. Moreover, we also analyze the potential interactions of the ligands and the atoms in the protein. This new data lets us compute interesting regions that are determined and labeled at the highest resolution level and hierarchically clustered for all the lower resolution levels. Finally, the user can explore the data using the charts or the 3D view.

3.1 Derived data calculation

From the input data we need to calculate two important quantities: the relative position of ligands (outside, on the surface, or inside the biomolecule) and their speed.

The **position** information is calculated using a two-step procedure. First, we heuristically determine whether the ligand is interacting with the protein using the geometric positions of their atoms. If that is not the case, the ligand is classified as *outside*. Otherwise, we further analyze whether the ligand is on the surface or inside the protein. In order to determine the interaction, we create a list L with all the atoms of the protein that are at a distance $d \leq r + 5\text{Å}$ of the ligand's centroid, where r is the circumradius of the ligand. The reasoning behind this heuristic is that, according to the domain experts, the energies determining docking between the protein and the ligand are dominated by close-range forces that decay over distance (e.g., van der Waals energies). However, this amount can be adjusted on a per-case basis if the simulation provides more information, such as the per-atom interaction energies. This heuristic is also used in the highlighting of the interactions of a ligand and the protein, and the 3D selection, explained in Sect. 3.4 and Sect. 3.5, respectively.

The position is then classified according to the following criterion:

$$position = \begin{cases} |L| \leq \lambda & \rightarrow outside \\ \lambda < |L| & \rightarrow on\ \ the\ \ surface\ \ or\ \ inside \end{cases}$$

The threshold λ is designed to avoid misclassifications in which the ligand travels close to too few atoms of the molecule for the exerted forces to be sufficient to "capture" the ligand. For our datasets, we empirically determined a suitable value of $\lambda = 5$. Like in the previous case, actual per-atom energies would open the possibility to design a data-guided parameter. To further distinguish between *surface* and *inside* in the latter case, we use a plane P centered at the centroid of the ligand, whose normal N is the vector that goes from the centroid of the ligand to the centroid of the protein. We then calculate how many atoms of the protein lie in each hemispace defined by the plane. If more than 70% of the atoms lie in the positive hemispace, then the ligand is considered to be *on the surface*. Otherwise, it is classified as *inside* the protein. This classification is depicted in Figure 2. The value of 70% was also determined experimentally. However, a more exact parameter can be calculated by analyzing all the cavities of the protein and evaluating how many atoms are necessary to consider the ligand inside, but our progressive exploration and coordinate 2D and 3D views make this unnecessary and even impractical for its prohibitive cost. We then classify different portions of the trajectory using this information. We use three different classes: *outside*, *on the surface*, and *inside*. Since bonding affinity is usually correlated with slow speeds, we have also analyzed the data in terms of speed —to build a second classification and clustering— but we found that the resulting clustering was almost equivalent to the one built from the position only. It makes sense, because the interaction forces are the ones that effectively prevent the ligand from escaping from the protein. So when the ligand is close, it is highly probable that different forces exerted between the protein and ligand will slow down the ligand's pace. As a consequence, we keep the position classification as the initial labeling by default. As described later, when more than one labeling is required, we add two widgets to change between them.

The **velocity** is computed as the difference in the position of the centroid of the ligand between consecutive snapshots. Since the simulation data contains outliers (a ligand may jump a large range from one frame to another, due to periodic boundary conditions or joining artifacts since some of the datasets are built by concatenating a small number of MD simulations), we analyze the data using Tukey's fences [16] with $K = 3$ for very large values. So, using quartiles 1 and 3, values over $Q3 + 3(Q3 - Q1)$ are identified and discarded. The speed is then normalized taking into account the maximum and minimum values over the trajectory, so that it can be plotted along with the distance in the same chart.

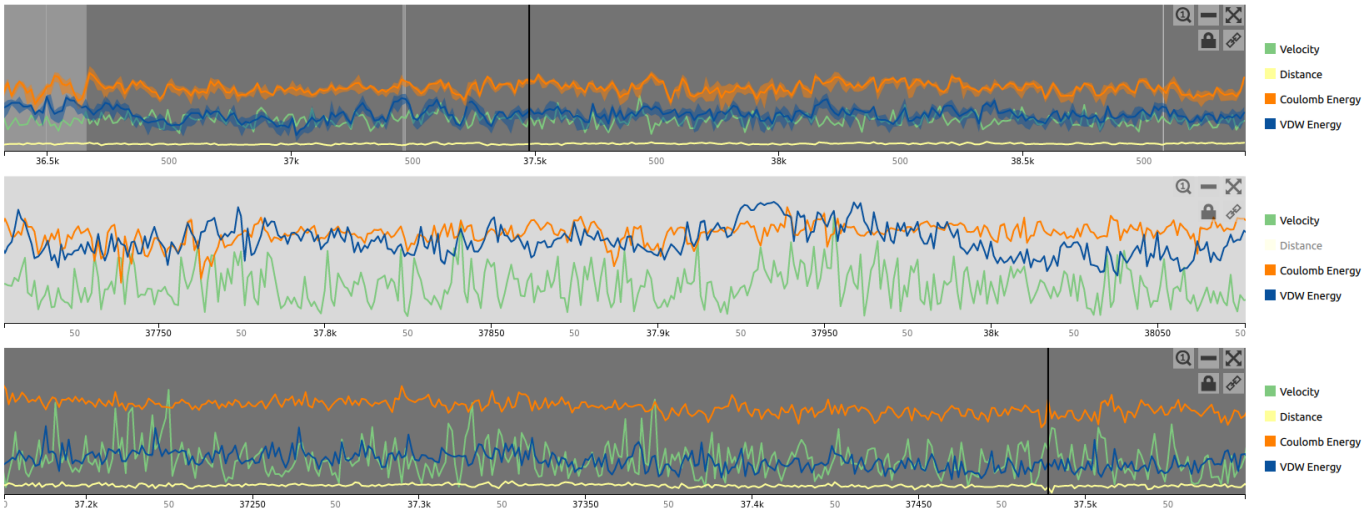


Fig. 3. Enhanced charts: The charts show four values: Coulomb energy, Van der Waals energy, distance, and speed. The background is color-coded with the active labeling. In this case, only one clustering has been calculated: the one that encodes the importance of the trajectory point based on the relative position of the ligand with respect to the protein. The darker the region, the smaller distance to the inner part of the protein.

Finally, we also calculate the interaction information of the ligand(s) with the protein, and this is used to identify the potential regions of interest of the trajectories.

Clustering calculation. Researchers are interested in regions where the protein and ligand(s) interact. Starting at the highest resolution of the simulation, each snapshot has been tagged as outside, on the surface, or inside for each ligand. Therefore, at this level we already have all frames classified as *no interaction* if the ligand is outside the biomolecule, *surface interaction* if the ligand is on the surface, or *inside interaction* if the ligand is inside the biomolecule. We then join consecutive steps with the same classification into intervals. These intervals are then grouped using a complete-linkage hierarchical agglomerative clustering (HAC) algorithm based on CLINK [10]. The dissimilarity metric used is the maximum temporal distance between the frames of the different clusters. This is equivalent to the size of the resulting cluster, without using the classes. Consequently, clusters have similar width, which facilitates the navigation. Clusters are then classified according to their predominant interaction into four classes: 1, no interaction; 2, a mixture with more snapshots of surface interaction than with inside interaction (i.e., predominantly surface interaction); 3, a mixture, but with more frames for which the ligand is inside; and lastly 4, in which all interaction happens inside the molecule.

Notice that we are restricted to join the neighboring clusters, because our data forms a linear sequence. Thus, the complexity of the clustering algorithm is $\mathcal{O}(n \log n)$ in time, instead of the general optimal $\mathcal{O}(n^2)$, and $\mathcal{O}(n)$ in memory, since the number of possible pairs is linear, instead of quadratic.

3.2 Simulation overview

The initial view of our application provides a general depiction of the simulation, with a 3D view of the protein and ligand(s) configuration on top, and a set of enhanced charts with the information about each of the ligands participating in the simulation at the bottom (see Fig. 1).

The main interactions in the 3D view are translation, rotation, zoom, and clipping, as well as the selection of 3D volumes, discussed in Sect. 3.5. The charts can be explored individually or coordinately. They show a high-level representation of the simulation enhanced with region marks, that use the simulation data to hint about potential regions of interest. The charts can be further explored by zooming-in, dragging, and so on. Different interaction tools are provided: clicking on a selected cluster, zoom-in/out with the mouse wheel, or jumping directly to a certain trajectory point at the maximum resolution. Moreover, the simulation can be run at various speeds in all the widgets at once, so one can see the 3D configuration together with a position marker in the charts. Next, we introduce in detail the different features of each of those elements.

3.3 Enhanced energy plots

The energy plots are intended to show how the MD simulation fares. However, due to the fact that our simulations have tens or hundreds of thousands of steps, they cannot fit in the screen. Therefore, we have designed a hierarchical exploration scheme that starts with the whole trajectory and lets the user progressively explore more detailed regions. Higher levels group tens or hundreds of steps in a small number of pixels. The main value we visualize is the average. However, the minimum and maximum values within those small ranges can vary greatly. To provide further insights, we also encode the energy variations displaying first and third quartiles as a shaded region with a less saturated color around the energy lines. This gives an overview of the data variations within the range.

With this visualization motifs as a basis, our system further improves the progressive exploration by providing two features:

- Visual encoding of the ligand behavior.
- Clickable elements for fast exploration.

In the following, we first describe the different visual elements designed to improve the communication of the data presented and then introduce different interaction techniques that help to quickly explore the simulations.

3.3.1 Visually encoding MD data

Since we want to deal with MD simulations of many thousands of steps, simply encoding the average values in a hierarchical way is not enough. Further insights are required to save the user from spending a lot of time zooming-in and out and dragging back and forth to get the details of the data. Our system analyzes the data and generates visual elements that provide clues on where to start the data exploration. Protein-ligand interactions occur when the ligand(s) are on the surface or inside the protein. Thus, we will enhance the energy depictions with information related to the positions of the ligands, since these, in combination with other information, such as ligand speed, may indicate potential regions of interest. All of this is encoded visually (see Fig. 3). In this way, the exploration of the data is greatly facilitated and accelerated.

In the previous section, we described how we cluster the different MD steps based on the atom positions and the derived information, and how we classify the resulting clusters into four different groups:

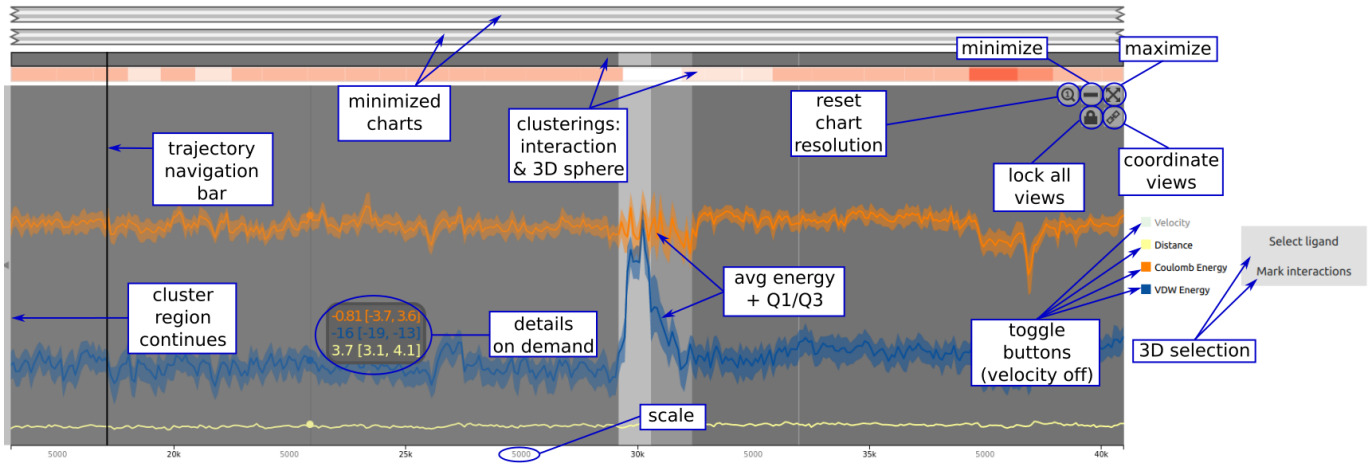


Fig. 4. The elements to interact with charts: The user can get the details-on-demand by hovering over the chart with the Ctrl button pressed. The charts can be minimized to give more room for other charts, and an icon simulating a folded paper provides a visual cue on the existence of other charts. Values can be toggled on/off with the right legends, that work as buttons. Moreover, the top right buttons perform different actions. From left to right and top to bottom: reset the view to maximum zoom-out level, minimize and maximize, lock all the charts to this one, and coordinate them to this level of zoom and position. Further operations are also possible, such as chart dragging, setting the trajectory current playing position to a certain one (Ctrl+left click) , maximize a user-defined region (right button + drag), ...

1, no interaction; 2, predominant surface interaction; 3, predominant interior interactions; and 4, only interior interactions. This information is communicated to the user by displaying it as the background color of the chart with rectangular areas that can be clicked on to magnify the corresponding region. We selected different saturations of gray to encode this information, since these color differences are still easily perceived by any user and they do not overlap with other visual cues of our system. Clusters of type 1 have white background. Clusters of type 2 are visualized with a low saturation gray background. Clusters of type 3 use a slightly more saturated shade of gray, and, lastly, clusters of type 4 receive the darkest shade of gray as background. By combining this information with the energy, we can spot sites with potential binding affinity, such as when the ligand is inside the protein and its speed is low. As we will see later, the same clustering strategy is also used for 3D to 2D interaction, and we may have more than one clustering at a given point in time.

3.3.2 Interacting with the chart

The chart widget provides several elements to facilitate data exploration (see Fig. 4). The rightmost legends are buttons that can toggle on/off the distance and velocity plots. For the energy values, the behavior is slightly different: since researchers consider it of utmost importance, when they are on, the energy lines are shown and the first quartiles around the average are also depicted. When toggled off, the line is de-emphasized and the Q1 and Q3 values disappear.

The first and most visible interaction elements in the chart appear in the top right corner. These five buttons trigger different actions:

Reset: The magnifying lens icon with a number one inside switches back to the initial size and position of charts.
Minimize/maximize: The charts can be individually minimized to provide more space for the other charts, by clicking the *minus* button. The maximization button minimizes the other charts (as shown in the top part of Fig. 4).
Lock all charts: All charts are set to the same zooming level and position as the one we have clicked on. By locking, further chart exploration (with any chart) is coordinated.
Coordinate charts: The link icon button sets the other charts to the same zoom level and position as the current one, but the coordination is not locked.

And the direct manipulation tasks allowed over the chart are:

Zoom to a cluster: The user may zoom in by clicking on the cluster region, which will magnify the selected region. Right-click will zoom back to the previous zooming level and position.
Continuous zooming-in/out: With the mouse wheel, the user can achieve detailed zooming.
Dragging: The chart can be dragged around with the left mouse button.
Center cluster: If a selected region (in a cluster) does not fit in the current view, arrows at the left or right part of the chart indicate that the region continues outside the chart. These arrows act as buttons that can be clicked to center the cluster on screen.
Explore range: Dragging with the right button lets the user define an arbitrary range, which is then magnified to fill the chart.
Detailed data: Control button opens the detail view, providing data on the actual values of the charts as the user hovers over them. The data displayed consists of: the encoded value, the average, and the minimum and maximum values in the range that is represented by the current pixel (see Fig. 4).
Set the navigation point: By Ctrl+click in any chart, the user can set the navigation step to be displayed in the 3D viewport.

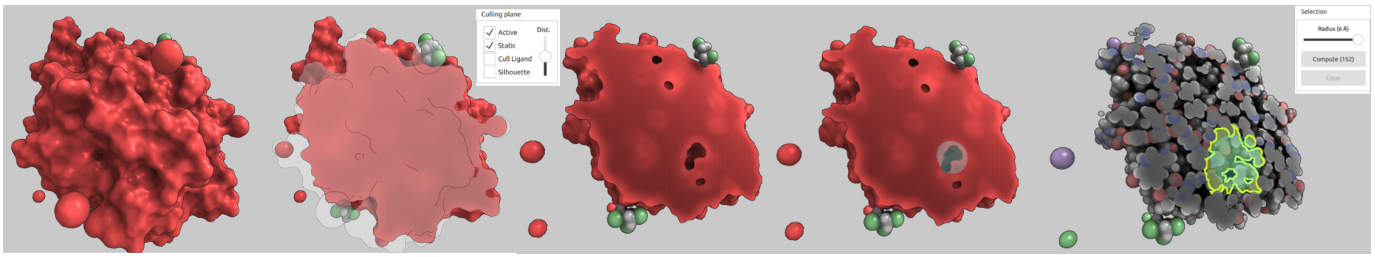
Additionally, the auxiliary variables (distance and speed) can be toggled on and off, and hovering over the chart highlights the class (cluster range) that would be selected with a left click.
Ligand interaction with the protein: The regions of interest are those where the ligand is inside the protein and interacting with its atoms. Thus, we generate a hierarchical labeling of the trajectory that encodes this information. It is shown as the color of the background of the chart.
Intersection with 3D volume: In order to facilitate the exploration from the 3D view to the charts, we let the user select a 3D region, and then mark on the charts where the ligands are placed with respect to the 3D selected volume.

The first classification, derived from the simulation data, allows the users to go from the plot to the potentially interesting situations of the simulation. If, on the contrary, the researchers already have a region of interest in the 3D simulation that they want to evaluate, they can use the secondary labeling strategy.

All the colors in the charts were selected among colors that contrast well to each other using the Color Brewer system.

3.4 3D exploration

The 3D exploration is a common molecular viewing technique that can represent the protein and the ligands with different motifs. Available rendering modes are: Van der Waals, balls and sticks, licorice, ribbons, and Solvent Excluded Surfaces (or SES [18]). Besides being able to see the trajectory in real-time (we read and render up to 60 steps of the trajectory per second), there are also other tools for visual exploration.



(a) Initial state (b) Clipping (c) Locate cavity (d) Set sphere (e) Adjust radius

Fig. 5. Interactive selection of a 3D region inside the protein with the help of a clipping plane. We set SES representation for easy cavity detection (a). Then, we start clipping (b) with the clipping plane tool (depicted between b and c). The silhouette of the clipped geometry can be removed to facilitate the search for cavities (c). Once the cavity is found, the user can interactively place a bounding sphere on the surface of the clipping plane (d). Finally, the radius of the sphere can be interactively changed (e). Throughout this process, the interface shows the number of affected atoms (*Compute* button, shown top right), and these are identified in the 3D view by highlighting them at the same time.

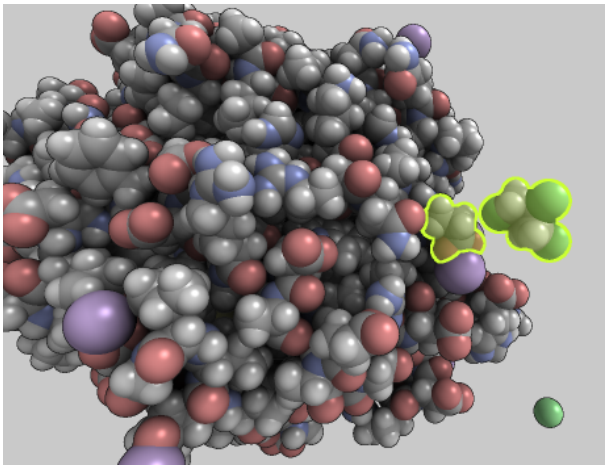


Fig. 6. Interactive highlighting of interacting atoms with the selected ligand. As the ligand moves, the atoms interacting with it are emphasized.

First, we can highlight the ligand of interest (buttons at the right of the chart). The highlighting consists in applying a yellow color at the silhouette of the ligand, and a semi-transparent yellow to the atoms of the ligand themselves. This makes it very easy to distinguish the selected element from the atoms of the protein (see Fig. 1).

Highlighting interactions. Researchers are very interested in understanding which atoms and residues are interacting with the ligand. Thus, we have added the highlighting also to these atoms. We determine which atoms are interacting by measuring the distance with respect to the ligand's centroid. This calculation is carried out in a pre-processing step, when the distances and velocities are calculated. In this pre-process, we store a list of interacting atoms for each ligand per step. In order to take advantage of the high temporal coherence that interactions exhibit, instead of storing a different list per frame, we store the intervals (first and last frame) at which the atoms interact with the ligand. Thus, for example, for ligand 1, we could have a list of interactions with each atom $(A_1, A_2, \ldots, A_n)$ similar to, e.g., this: $A_1 = \{[0,23],[60,80]\}, A_2 = \{[4,8]\}, A_3 = \{[4,70]\}, A_4 = \emptyset$.

These lists are used to highlight the atoms that interact with the ligand. An example of how these atoms are interactively emphasized is shown in the accompanying material. Fig. 6 shows the result in an image. At each frame, we use these lists to check for each atom whether it is interacting or not, at that point, with the ligand. This is done using a dicotomic search, so the total cost is $\mathcal{O}(n \log k)$ where n is the total number of atoms of the biomolecule, and k is the size of the longest list of interaction intervals.

Clipping plane. We can also partially or completely clip the protein (and ligands, if desired) with a clipping plane. This is especially useful to inspect in detail the regions around the ligand when it is close to a binding position. Since binding commonly happens in cavities, using the clipping plane allows us to identify the interesting residues that are interacting around the ligand. The clipping plane is also a key element for visually identifying cavities inside a protein. By combining SES and plane clipping, the researchers can quickly locate regions of interest that can be eventually selected to analyze the interactions with the ligand if required, as explained next.

3.5 3D selection

Besides being able to cope with very large simulation paths, another feature that makes our visualization system stand out with respect to other trajectory exploring packages is the ability to go from the 3D view to the 2D plots. Traditional software packages only provide the 2D to 3D step, that is, the user inputs happen mainly using 2D tools, and the 3D view is used as inspection of a given step. With some exceptions (e.g., [15]), the trajectory cannot be explored back and forth around a certain snapshot. We provide means to go from the 3D to the 2D by letting the user explore and select volumetric regions using the 3D viewing widget. Users can select cavities and get immediate feedback on whether these cavities were visited by any of the ligands along the simulation, and see when that happens.

The selection procedure works as follows:

1. Clip the geometry using the clipping plane.
2. Place an influence sphere on the surface of the clipping plane.
3. Edit the radius of influence.
4. Confirm the volume.

The clipping plane described in the previous section is also used as a support for the volume selection. In the first step, the clipping plane is used to access the interior of the protein. Here, the SES representation is the most suitable thanks to its inherent ability to show the cavities of the protein, cf. Fig. 5-(c). Once the plane has been set, the user can place a sphere on it, by Shift-clicking; the sphere then appears and follows the mouse movements (step 2, illustrated in Fig. 5-(d)). When the user is satisfied with its position, he or she fixes it with a left click. Next, the radius of the sphere can be changed with a slider or with the mouse wheel (third step), and the affected atoms are highlighted accordingly, as shown in Fig. 5-(e). When the user presses the *Compute* button, the trajectories will be annotated to indicate when each ligand interacts with the selected atoms. This is done using the lists of atom interactions mentioned earlier. Given the list of atoms within the influence sphere, we run a 1D sweep line algorithm on the lists of atom interactions to find the ranges of snapshots where these atoms actually have an interaction, and categorize these ranges according to the relative importance of the interaction. For example, assume that A_1, A_2, and A_3 in the example lists of Sect. 3.4 are the three atoms selected by the influence sphere. Then given those example

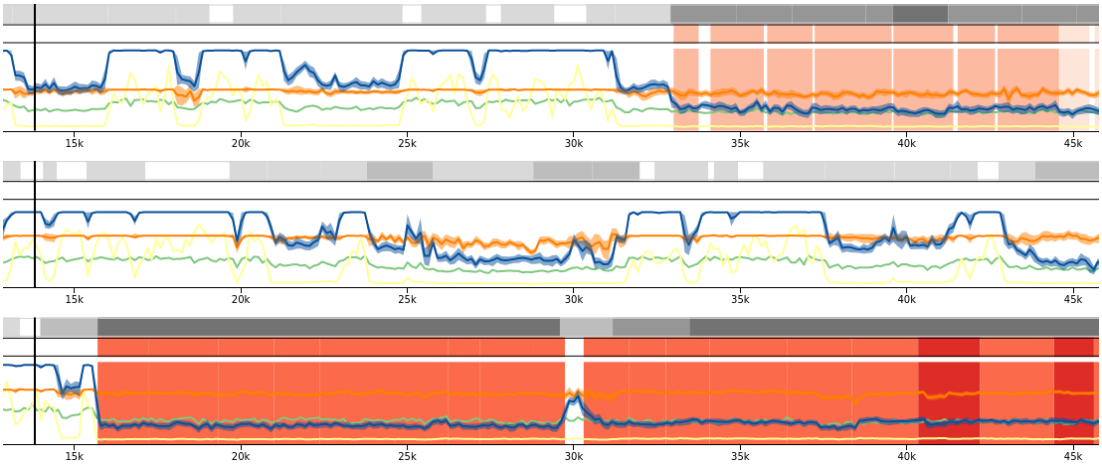


Fig. 7. Color coding the occupancy of 3D volume by the ligands. After the volume has been confirmed, the system provides visual feedback on the interaction of the ligands with the selected volume. The new color coding (in reddish tones) is now added to the top of the chart, together with another bar indicating the position classification.

lists, the 1D sweep line algorithm would yield: $R = \{[0,4,|\{A_1\}| = 1],[4,8,|\{A_1,A_2,A_3\}| = 3],[8,23,|\{A1,A3\}| = 2],[23,60,|\{A_3\}| = 1],[70,80,|\{A_1\}| = 1]\}$, and since we are not interested here in the identity of the particular atoms interacting, we can store this more compactly as $R = \{[0,4,1],[4,8,3],[8,23,2],[23,60,1],[70,80,1]\}$. Each triplet is in the form of $[a,b,w]$, indicating that w of the selected atoms interact with the ligand in all snapshots in the interval $[a,b]$. These numbers w are then normalized by dividing them into the total number of atoms selected by the influence sphere (in this example 3), producing a weight $\bar{w} \in [0,1]$. We then classify intervals into four categories: *no interaction* (when $\bar{w} \in [0,0.2)$), *low-medium interaction* (when $\bar{w} \in [0.2,0.6)$), *medium interaction* (when $\bar{w} \in [0.6,0.8)$) and *high interaction* (when $\bar{w} \in [0.8,1.0]$). We label these regions using a scale of low to high saturated reddish backgrounds, and compute a second hierarchical clustering for this criterion (see Fig. 7). Note that, previous to this clustering definition, only one was present, so the only visual representation was the one in the background of the chart. Now, with two clusterings present, we need to provide a tool for the user to change between them. We do this in the form of bars placed on top of the charts, that can be clicked to activate the corresponding clustering, but are also color coded with the values of the range classifications. In this way, even if a certain classification is not active, the user still has the visual information close to the chart, and in correspondence with it, he or she can visually compare all the clusterings that have been computed, as shown in Fig.8. This new clustering not only allows the users to quickly find out if any of the ligands visit the selected region, but also how strong the interaction is along the trajectory. As before, the information is computed at maximum resolution, and clustered hierarchically. When the user navigates to the steps of the trajectory where these interactions happen, the interacting atoms can be highlighted using the rightmost button labeled *Mark interactions*, and the interacting atoms will be emphasized. The result can be seen in Fig. 9.

3.6 Coordinated views

The 3D and 2D views are coordinated: the user may define the trajectory snapshot and the 3D view will change accordingly. Moreover, actions over the 3D view, such as volume selection, will also update the 2D charts. Moreover, the multiple charts can also be explored coordinately or independently. We found it useful to have two levels of coordination: *i)* fixing views, and *ii)* locking coordination. In the first case, when exploring a chart, the user may bring the other charts to the same zooming level and position. In the second case, not only the charts are set with the same configuration, but subsequent changes (zoom in/out, reset, drag, etc.) affect all of them in the same way.

For the purpose of the interactive data exploration, it is crucial to

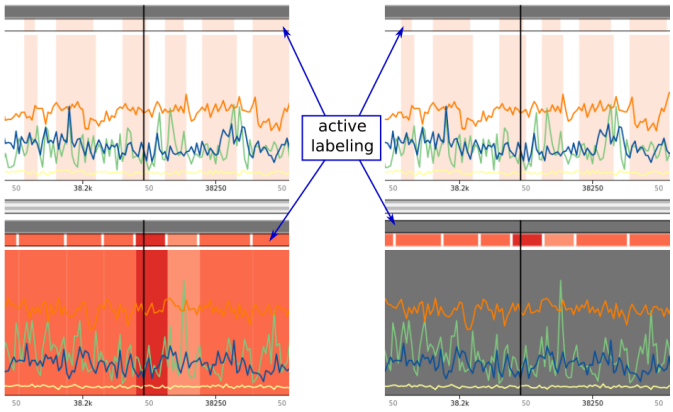


Fig. 8. Multiple trajectory classifications can be handled at once. The top line is a widget that allows changing between labeling visualizations. The left image shows the cavity proximity labeling, while in the right image, the bottom chart shows the interaction labeling.

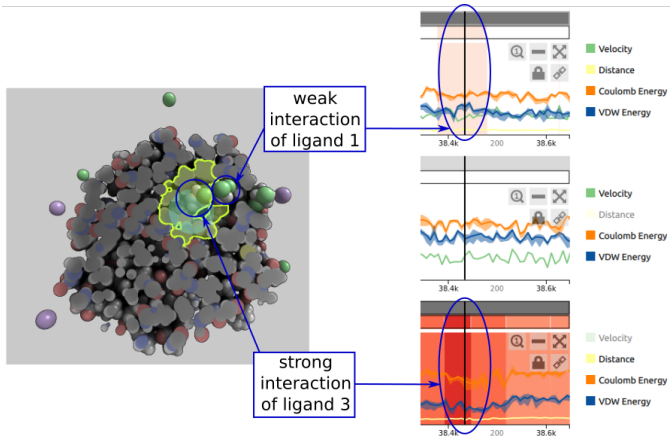


Fig. 9. After a volume has been selected and the clustering updated, if we navigate to the trajectory points where higher interaction is detected, we will find the ligand involved in the selected cavity region. Note that in this example, a second ligand also has a weak interaction, as it is located very close to the selected spherical region.

have both possibilities: coordinating charts and decoupling their interaction. Since different ligands may interact with the protein at different moments of the trajectory, decoupled charts let the user compare the outcomes of different snapshots by zooming on them individually, and observing them side by side. On the other hand, setting all the views with the same zoom level and position is also necessary, because at the most detailed views the user would require lengthy dragging operations to set all charts at the same position. Fully coordinated exploration is also useful in cases as the one illustrated in Fig. 9, where two ligands are interacting with the same cavity at the same time. The same can be said of the 2D to 3D operations: since the trajectories are so long, playing them continuously makes no sense, even with 60 snapshots per second, which is our default playback speed, it would require almost 14 minutes to play the whole sequence. For longer trajectories, the situation is even worse.

4 Results

The presented visualization system is able to quickly load the data and present it to the user. The initial view is an overview that shows insights to foster the user's *informed* exploration of the detail data. Since the datasets consist of thousands of steps, the overview lacks enough details and low-level inspection is required. The initial view provides a lot of data-derived cues that help the user quickly deduce features, such as whether any ligand seems to have bound the molecule or which are the simulation ranges that appear to have interesting interactions.

4.1 Use cases

Our system can be used in several ways. To demonstrate its potential, we discuss two example use cases within this section.

Classical 2D to 3D exploration. The workflow that domain experts use consists on first analyzing the 2D charts and then inspect the 3D view. With our system, this can be accelerated because we show thousands of steps at a glance in a 2D plot, and the inspection tools facilitate drilling down to the details quickly. Moreover, each chart can be inspected individually, if there are portions of the trajectory that are interesting for the different ligands, or jointly, if the user wants to analyze a certain part. For example, analyzing the final stages of the simulation can be done by: *i)* locking all the charts, *ii)* zooming with the wheel in any of those (or by clicking on the labeled regions if any is available), *iii)* Ctrl+click at the end of the chart, to set the 3D view at the desired position. This procedure can be used to analyze the outcome of a simulation, when the researcher knows what to search for, and the answer can be obtained in few seconds, for all the ligands at once.

Exploratory 3D to 2D. The proposed visualizations can also be used in an exploratory fashion, especially when exploiting the linking from 3D to 2D. Since the researchers usually have a priori knowledge on what the active sites of the molecule are, they can mark these sites to query whether these were visited by the ligand(s). The procedure is simple, and has been described in Sect. 3.5. To sum-up, it would consist of: *i)* clip the geometry searching for a cavity, *ii)* mark the volumetric region, and *iii)* confirm the selection. Throughout the entire process, the user has visual cues that help to properly select the 3D point, since the cavity can be properly seen if the SES representation is used, and the range of influence is also visually depicted. Then, by looking at the chart now communicating the results of the query, the user can quickly grasp which (if any) and where the ligands interacted with the marked 3D volume. This is a novel feature enabling new ways of exploring molecular simulations.

4.2 System performance

In this section, we would like to briefly summarize the performance with which our system runs. Our system is a Qt application that uses an OpenGL window for the molecular 3D visualization and a web browser, more concretely, *QtWebEngine*, an engine based on Chromium, for the chart depiction. These charts are drawn using JavaScript and the D3 library in the browser, and the communications with the rest of the Qt application are handled using the *QtWebEngine*. The 3D rendering supports many molecular representations including Solvent Excluded Surfaces, and a fast implementation of object-space ambient occlusion.

Within the proposed system, the simulation data is handled in the AMBER format [33] and we use the tools provided by the AMBER library to process this data. In order to manage the data efficiently, our system preprocesses the input data. First, the whole trajectory is read and the speed and position charts are computed, along with the lists of interacting atoms. Upon application start, the first 3D configuration is rendered immediately, and the data processing starts. After that (which may take up to one or two seconds) chart widgets are built and rendered. In total, the loading tasks are completed in less than three seconds for a model containing three ligands. The 3D configuration is then loaded and displayed in less than one second, and the charts are rendered progressively, with every chart requiring less than one second to appear. Recomputing a labeling can be done interactively, as demonstrated in the video. A simulation with 800K snapshots takes also similar time. Note that we do not account for the preprocessing time here, which grows approximately linearly with the number of trajectory snapshots.

4.3 Evaluation

To obtain expert feedback, we conducted a demo session with informal feedback as well as a structured questionnaire. The participants of this study were six experts in protein engineering, working with MD simulations on a daily basis. This group consisted of one senior researcher (group leader), three post-doc researchers, and two PhD students. The demo session went as follows. There was a presentation of the tool, introducing different features of the application. After that, experts asked several questions on the tool. This initiated an informal discussion when the domain experts suggested some lines for future enhancements of the system. At the end, they were given a questionnaire to evaluate the visual cues and how the different features provided by the tool could help them in their daily work. The questionnaire also provided the space for suggesting possible improvements.

The questionnaire consisted of three groups of questions. The first one was on perceived usefulness of the system, the second one was on perceived ease of use, and the last one asked about the specific features of the system. In each of these groups, we had four to six more specific questions and the experts were asked to rank them on a scale between 1 (completely disagree) to 5 (completely agree).

The questionnaire revealed that the participants confirmed our expectation that our tool can be useful for their job (average 4 out of 5). Concerning the ease of use, all of them agreed that the application was easy to learn with average marks of 3.8 out of 5 on all the questions related to this group.

Concerning the features specifically designed for the processing of large simulations, the users confirmed that they were of high utility for them. When asked whether they thought the visualization of multiple ligands could be useful for their work, the average of answers was 4.5 out of 5. They also highly appreciated the gray encoding of the trajectory clusters, since they comprehensibly communicate the candidate areas for further exploration. In this case, the value was 4.8 out of 5. They also found the highlighting of atoms close to the ligand very useful to infer the ligand behavior (4.6 out of 5). Finally, the 3D selection of cavities was deemed useful, but not as importantly as the previous ones (3.6 out of 5). However, one of the suggestions of the experts was specifically in this area. They suggested us to add the possibility of determining the cavity position by using the 3D coordinates (that are commonly obtained using other cavity analysis software).

5 Discussion

Within this section, we discuss the advantages and limitations of our presented visual analysis system.

5.1 Advantages

The main advantages of our system are twofold. First, it enables users to visualize and interactively explore very long trajectories. We have demonstrated this with the example of an MD simulation of 50K steps, though larger simulations are also possible to load and explore. The second advantage lies in the proposed interaction capabilities.

For the progressive inspection of large-scale MD simulations, our system provides new features in the form of *enhanced charts*. We

compute the derived information such as position, speed, and ligand interactions with the protein atoms. As a result, we can add more meaningful information to the charts and overlay the interaction information in the form of clickable widgets that accelerate the user interaction. The regions that indicate the amount of interaction between the ligand and protein facilitate the progressive exploration, since their selection produces the magnification of an entire range of data. Thereby, the user is guided to the potential regions of interest, that can be accessed quickly, by means of a simple click. The domain experts appreciated also the speed of our system. Loading a trajectory and computing the derived information takes only several seconds, even for very long simulations. This is a significant improvement in comparison with the existing tools processing MD simulations, where such a task takes up to dozens of minutes.

Another way in which our system is unique is that it provides a new way to interact with molecules. Most existing visualization approaches let the user first explore the 2D energy charts before they guide her to a 3D view to further make sense of the simulation configuration. Even many commercial packages do not let the user go back and forth within the simulation, and only show a single step (e.g., in Maestro [28] or VMD [17]). In contrast, our system provides this method of data exploration. Moreover, in 3D it also adds highlighting the contacting atoms, which is a very appreciated addition. We also provide the inverse work-flow: the users may start with the 3D visualization and select a volume of interest to update the 2D charts, showing the points in the simulation where this volume is interacted with, and to what extent. This, besides its novelty, is of great utility, since the researchers do not need to interactively inspect all the trajectories, which may be tedious and time consuming. Instead, the information regarding if, which, and how many ligands visited a certain 3D region can be obtained immediately.

5.2 Limitations

The system has been built to represent at most three ligands' energy charts, and up to four data plots per chart for very long trajectories. We have tested with simulations of up to 800K snapshots, and the system had no problems with such amount of data. However, if more than three ligands at the same time should to be shown, it would be difficult to do so on the screen with the current representation, without removing the 3D view at least temporally. In spite of that, most MD simulations deal with a single ligand, so the space available for the 3D visualization is even larger than the one shown in the images across the paper.

The clustering algorithm is fast and can be easily adapted to different data. On the other hand, if many clusterings are required, they would also consume some screen real estate on top of the charts, and the space will be diminished.

Another current limitation is the number of different highlights that can be applied in the 3D view. Currently we cannot emphasize with different colors all the ligands present in the simulation.

5.3 Comparison with other software tools

As already stated, our system provides several advantages over other widely-used packages, with respect to the task of exploration of MD simulations. For instance, the way to analyze multiple ligand-protein interactions in LigPlot+ [21] consists of a set of 2D planar maps of the 3D configuration. It may, however, output a 3D configuration that can be viewed using other programs, such as Pymol. On the other hand, it does not provide specific tools for progressive exploration of large MD trajectories. Schrödinger's SID [29] generates a set of static charts that can be written in PNG or SVG formats, but no interactive exploration is provided, nor 3D exploration of multiple snapshots. VMD [17], another popular program, is mainly devoted to the visualization of large molecular complexes and the analysis of MD trajectories. However, it does not contain the integrated layout we have, and it does not contain the 2D to 3D and 3D to 2D bindings we provide. Similar to VMD, PLIP [35] is focused on the analysis of Protein-ligand interactions, unfortunately, it does not deal with precomputed trajectories, and does not allow the exploration of the MD results. It works as a web service that can read entries from the Protein Data Bank. PyMol [36] is an open source package distributed and maintained by Scrhödinger whose objective is to render and animate 3D structures, not to perform the analysis of MD trajectories. TAMD proposes a similar dashboard view, but its widgets and interaction possibilities are limited [23]. Other packages perform the analysis of the trajectories only by extracting information from the simulation, not by providing a unified system for 2D and 3D analysis [34].

5.4 Lessons learned

From the discussion with the domain experts and questionnaires several improvements emerged, with a focus on its potential usability for the community. Among these, the selection of a cavity by stating its 3D coordinates or changing the parameters used as thresholds, especially the distance of 5Å used for determining the interactions with close atoms. These features can be easily added to the system.

The comparison between three ligand trajectories immediately led the biochemists to the request to compare several trajectories of the same ligand. This is definitely a very interesting future extension of the system which will require changes in our design decisions and maybe even adding yet another visual representation, as the number of such trajectories can be large (currently up to hundreds).

As the possibilities for MD simulation exploration by the currently available tools are very limited, the biochemists nowadays are forced to switch from detailed exploration to abstract graph representations, showing calculated energies, clusters according to different properties, etc. However, this can lead to omitting important parts of the simulation. We believe that our tool will enable them to efficiently combine and fully exploit the benefits of these two approaches.

6 Conclusions and Future Work

We have designed and implemented a visualization system built specifically to deal with very long MD trajectories. To facilitate the exploration of these long data sequences, we provide unique methods of two kinds: first, we calculate and visually encode information that may guide the user to regions where interesting interactions between the ligands and protein occur. This is performed by a data-based classification of the steps, and a hierarchical clustering that facilitates the representation of large portions of the trajectory at higher levels. Second, we provide a set of interaction tools to ease the quick exploration of the charts, as well as some novel interactions that permit manipulating 3D views and seeing the results in the 2D charts. Notably, the charts exploration is greatly facilitated by our hierarchical clustering algorithm which provides an adaptive, multi-scale navigation of the data. This technique can potentially be applied to other time-dependent data. To the best of our knowledge, these features are unique and different to the features provided by commercial software. Traditional packages do not supply tools for the integral exploration of such data sets, for instance the hierarchical, progressive exploration of charts is lacking. Furthermore, the common operations in 3D are labeling, distance calculation, and so on, but no bindings exist that facilitate the labeling of 2D charts based on 3D user input.

One of the potential extensions of our system would be the ability to load distinct simulations and add the 3D to 2D inspection in all of them at once. This is not straightforward, because in each simulation, the positions of the atoms change. Thus, some work is necessary to facilitate the exploration in a single 3D view and translate the interactions to all the other simulations. Another extension may be to incorporate multiple 3D views. With the current modular design, it should not be complicated, but this has not been suggested to us as a desirable feature, perhaps because the detailed inspection in 3D requires relatively large room and thus, even visual comparison may be difficult.

Acknowledgments

This work was supported in part by project TIN2017-88515-C2-1-R (GEN3DLIVE), from the *Spanish Ministerio de Economa y Competitividad*, by 839 FEDER (EU) funds, the Deutsche Forschungsgemeinschaft (DFG) under grant RO 3408/2-1 (ProLint), and the Czech Science Foundation (GACR) under grant GC18-18647J. We also would like to thank the reviewers for their valuable comments.

REFERENCES

[1] N. Alharbi, M. Alharbi, X. Martinez, M. Krone, A. Rose, M. Baaden, R. S. Laramee, and M. Chavent. Molecular visualization of computational biology data: A survey of surveys. *EuroVis short papers*, pp. 133–137, 2017.

[2] J. Byska, A. Jurcik, M. E. Gröller, I. Viola, and B. Kozlikova. MoleCollar and tunnel heat map visualizations for conveying spatio-temporo-chemical properties across and along protein voids. *Computer Graphics Forum*, 3(34):1–10, 2015. EuroVis 2015 - Conference Proceedings.

[3] J. Byska, M. Le Muzic, M. E. Gröller, I. Viola, and B. Kozlikova. AnimoAminoMiner: Exploration of protein tunnels and their properties in molecular dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):747–756, 2016.

[4] M. Chavent, B. Lévy, M. Krone, K. Bidmon, J.-P. Nominé, T. Ertl, and M. Baaden. GPU-powered tools boost molecular visualization. *Briefings in Bioinformatics*, 12(6):689–701, 2011.

[5] T. E. Cheatham III and D. R. Roe. The impact of heterogeneous computing on workflows for biomolecular simulation and analysis. *Computing in Science & Engineering*, 17(2):30–39, 2015.

[6] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015.

[7] J. D. Chodera and F. Noé. Markov state models of biomolecular conformational dynamics. *Current Opinion in Structural Biology*, 25:135–144, 2014.

[8] G. Ciccotti, M. Ferrario, and C. Schuette. Molecular dynamics simulation. *Entropy*, 16:233, 2014.

[9] E. Cuenca, A. Sallaberry, F. Y. Wang, and P. Poncelet. Multistream: A multiresolution streamgraph approach to explore hierarchical time series. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[10] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

[11] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer. Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1636–1649, 2017.

[12] K. Furmanova, M. Jaresova, J. Byska, A. Jurcik, J. Parulek, H. Hauser, and B. Kozlikova. Interactive exploration of ligand transportation through protein tunnels. *BMC Bioinformatics*, 18(2):22, 2017.

[13] P. Godfrey, J. Gryz, and P. Lasek. Interactive visualization of large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2142–2157, 2016.

[14] S. Grottel, M. Krone, C. Müller, G. Reina, and T. Ertl. Megamol – a prototyping framework for particle-based visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 21(2):201–214, Feb 2015.

[15] P. Hermosilla, J. Estrada, V. Guallar, T. Ropinski, A. Vinacua, and P.-P. Vázquez. Physics-based visual characterization of molecular interaction forces. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):731–740, 2017.

[16] D. C. Hoaglin. John W. Tukey and data analysis. *Statistical Science*, pp. 311–318, 2003.

[17] W. Humphrey, A. Dalke, and K. Schulten. VMD: visual molecular dynamics. *Journal of molecular graphics*, 14(1):33–38, 1996.

[18] B. Kozlikova, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures: State of the art revisited. In *Computer Graphics Forum*, vol. 36, pp. 178–204. Wiley Online Library, 2017.

[19] M. Krone, B. Kozlikova, N. Lindow, M. Baaden, D. Baum, J. Parulek, H.-C. Hege, and I. Viola. Visual analysis of biomolecular cavities: State of the art. In *Computer Graphics Forum*, vol. 35, pp. 527–551. Wiley Online Library, 2016.

[20] T. J. Lane, D. Shukla, K. A. Beauchamp, and V. S. Pande. To milliseconds and beyond: challenges in the simulation of protein folding. *Current Opinion in Structural Biology*, 23(1):58–65, 2013.

[21] R. A. Laskowski and M. B. Swindells. LigPlot+: multiple ligand–protein interaction diagrams for drug discovery. *Journal of Chemical Information and Modeling*, 51(10):2778–86, 2011.

[22] M. Le Muzic, L. Autin, J. Parulek, and I. Viola. cellVIEW: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In *VCBM*, pp. 61–70, 2015.

[23] I. Likhachev, N. Balabaev, and O. Galzitskaya. Available instruments for analyzing molecular dynamics trajectories. *The Open Biochemistry Journal*, 10:1, 2016.

[24] N. Lindow, D. Baum, A.-N. Bondar, and H.-C. Hege. Dynamic channels in biomolecular systems: Path analysis and visualization. In *Proceedings of the 2012 IEEE Symposium on Biological Data Visualization (BioVis)*, BIOVIS '12, pp. 99–106. IEEE Computer Society, 2012.

[25] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum*, vol. 36, pp. 527–538. Wiley Online Library, 2017.

[26] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):321–330, 2017.

[27] T. Luciani, J. Wenskovitch, K. Chen, D. Koes, T. Travers, and G. E. Marai. FixingTIM: interactive exploration of sequence and structural data to identify functional mutations in protein families. In *BMC proceedings*, vol. 8, p. S3. BioMed Central, 2014.

[28] Maestro, Schrödinger, LLC. Schrödinger release 2018-1. https://www.schrodinger.com/maestro, 2018.

[29] Maestro, Schrödinger, LLC. Schrödinger release 2018-1: Desmond molecular dynamics system, 2018. Maestro-Desmond Interoperability Tools, Schrödinger.

[30] S. Malik, B. Shneiderman, F. Du, C. Plaisant, and M. Bjarnadottir. High-volume hypothesis testing: Systematic exploration of event sequence comparisons. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(1):9, 2016.

[31] J. D. Mercer, B. Pandian, A. Lex, N. Bonneel, and H. Pfister. Mu-8: visualizing differences between proteins and their families. *BMC Proceedings*, 8(Suppl 2):S5, Aug. 2014. doi: 10.1186/1753-6561-8-S2-S5

[32] S. I. O'Donoghue, K. S. Sabir, M. Kalemanov, C. Stolte, B. Wellmann, V. Ho, M. Roos, N. Perdigao, F. A. Buske, J. Heinrich, et al. Aquaria: simplifying discovery and insight from protein structures. *Nature Methods*, 12(2):98, 2015.

[33] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, 91(1-3):1–41, 1995.

[34] D. R. Roe and T. E. Cheatham III. PTRAJ and CPPTRAJ: software for processing and analysis of molecular dynamics trajectory data. *Journal of Chemical Theory and Computation*, 9(7):3084–3095, 2013.

[35] S. Salentin, S. Schreiber, V. J. Haupt, M. F. Adasme, and M. Schroeder. PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Research*, 43(W1):W443–W447, 2015.

[36] D. Seeliger and B. L. de Groot. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *Journal of Computer-Aided Molecular Design*, 24(5):417–422, 2010.

[37] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91–97, 2008.

[38] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, 2014.

[39] W. F. van Gunsteren and H. J. Berendsen. Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry. *Angewandte Chemie International Edition*, 29(9):992–1023, 1990.

[40] M. C. Watson. Time maps: A tool for visualizing many discrete events across multiple timescales. In *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 793–800. IEEE, 2015.

[41] K. Wongsuphasawat and D. Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2659–2668, 2012.

[42] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1747–1756. ACM, 2011.

[43] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 259–268. ACM, 2015.